

MODUL PRAKTIKUM VII

POLYMORPHISM

A. Tujuan

Mahasiswa mampu menerapkan dan menganalisa konsep *polymorphism* dalam Java

B. Latihan

Latihan 1:

Implementasikan *source code* berikut :

1. Base.java

```
public class Base {  
    private void amethod(int iBase) {  
        System.out.println("Base.amethod");  
    }  
}
```

2. Over.java

```
public class Over extends Base {  
    public static void main(String argv[]) {  
        Over o = new Over();  
        int iBase = 0;  
        o.amethod(iBase);  
    }  
    public void amethod(int iOver) {  
        System.out.println("Over.amethod");  
    }  
}
```

Output dari program di atas adalah:

Latihan 2:

Implementasikan *source code* berikut :

1. MyParent.java

```
public class MyParent {  
    int x, y;  
    MyParent(int x, int y) {  
        this.x = x;  
        this.y = y;  
    }  
    public int addMe(int x, int y) {  
        return this.x + x + y + this.y;  
    }  
    public int addMe(MyParent myPar) {  
        return addMe(myPar.x, myPar.y);  
    }  
}
```

2. MyChild.java

```
public class MyChild extends MyParent {
    int z;
    MyChild(int x, int y, int z) {
        super(x, y);
        this.z = z;
    }
    public int addMe(int x, int y, int z) {
        return this.x + x + this.y + y + this.z + z;
    }
    public int addMe(MyChild myChi) {
        return addMe(myChi.x, myChi.y, myChi.z);
    }
    public int addMe(int x, int y) {
        return this.x + x + this.y + y;
    }
}
```

3. MySomeone.java

```
public class MySomeone {
    public static void main(String args[])
    {
        MyChild myChi = new MyChild(10, 20, 30);
        MyParent myPar = new MyParent(10, 20);
        int x = myChi.addMe(10, 20, 30);
        int y = myChi.addMe(myChi);
        int z = myPar.addMe(myPar);
        System.out.println(x + y + z);
    }
}
```

Output dari program di atas adalah:

Latihan 3:

Implementasikan *source code* berikut :

MyClass.java

```
public class MyClass {
    void myMethod(int i) {
        System.out.println("int version");
    }

    void myMethod(String s) {
        System.out.println("String version");
    }

    public static void main(String args[]) {
        MyClass obj = new MyClass();
        char ch = 'c';
        obj.myMethod(ch);
    }
}
```

Output dari program di atas adalah:

Latihan 4:

Implementasikan *source code* berikut :

1. Base.java

```
public class Base {
    int i = 99;
    Base() {
        amethod();
    }
    public void amethod() {
        System.out.println("Base.amethod()");
    }
}
```

2. Derived.java

```
public class Derived extends Base {
    int i = -1;
    public static void main(String argv[]) {
        Base b = new Derived();
        System.out.println(b.i);
        b.amethod();
    }
    public void amethod() {
        System.out.println("Derived.amethod()");
    }
}
```

Output dari program di atas adalah:

Latihan 5:

Implementasikan *source code* berikut :

1. Parent.java

```
public class Parent {
    public void method1() {
        System.out.println("Parent's method1()");
    }
    public void method2() {
        System.out.println("Parent's method2()");
        method1();
    }
}
```

2. Child.java

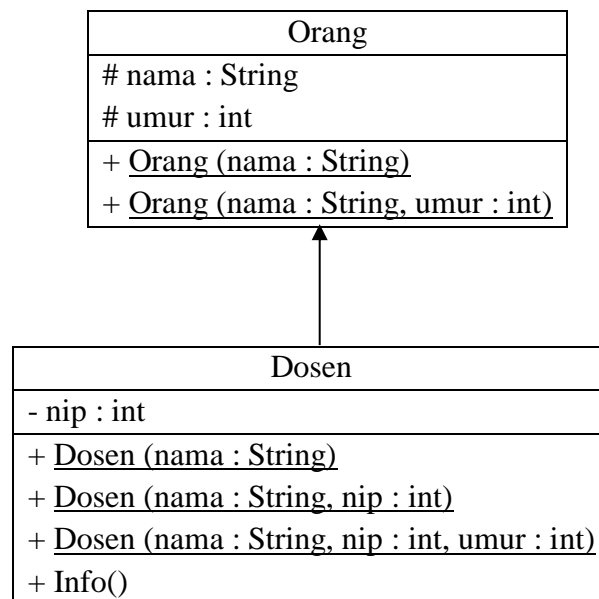
```
public class Child extends Parent {  
    public void method1() {  
        System.out.println("Child's method1()");  
    }  
  
    public static void main(String args[]) {  
        Parent p = new Child();  
        p.method2();  
    }  
}
```

a. Output dari program di atas adalah:

b. Bagaimana output dari program di atas apabila *syntax* `p.method2()` pada baris ke-8 Child.java diganti menjadi `p.method1()`?

C. Tugas Praktikum

1. Implementasikan UML *class diagram* berikut dalam program Java



Selanjutnya buatlah main class untuk mengetes program yang telah dibuat sebelumnya:

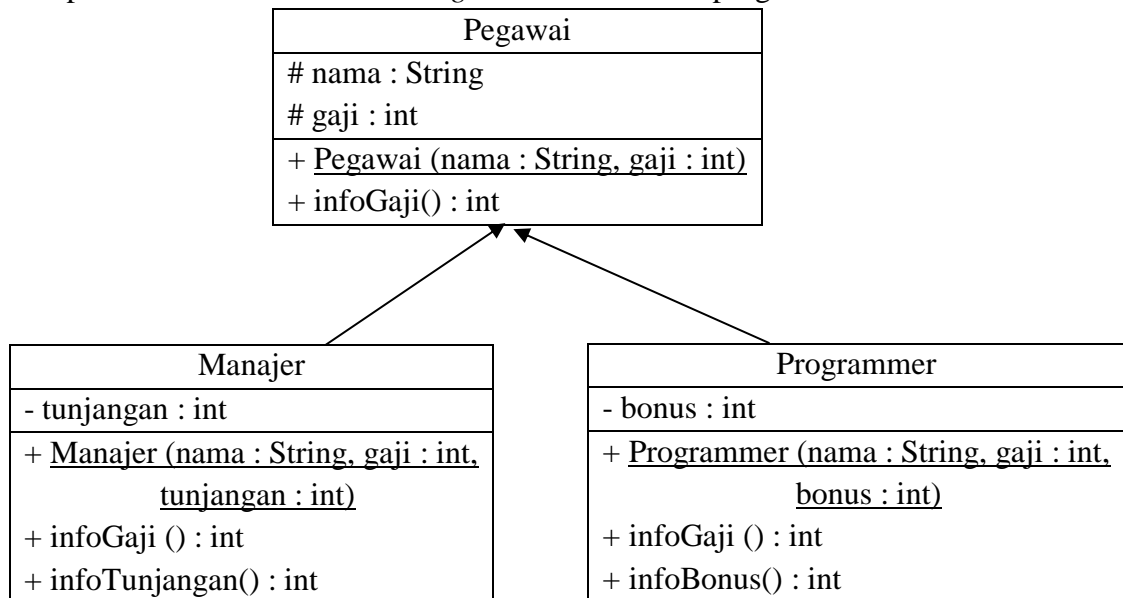
```
public static void main(String[] args) {
    System.out.println("Memasukkan identitas dosen 1 : Agus");
    Dosen dosen1 = new Dosen("Agus");

    System.out.println("Memasukkan identitas dosen 2 : Budi, NIP. 1458");
    Dosen dosen2 = new Dosen("Budi", 1458);

    System.out.println("Memasukkan identitas dosen 3 : Iwan, NIP. 1215, "
        + "umur 47");
    Dosen dosen3 = new Dosen("Iwan", 1215, 47);

    System.out.println();
    dosen1.Info();
    System.out.println();
    dosen2.Info();
    System.out.println();
    dosen3.Info();
}
```

2. Implementasikan UML *class diagram* berikut dalam program Java



Selanjutnya buatlah main class untuk mengetes program yang telah dibuat sebelumnya:

```

public class Bayaran {
    public int hitungbayaran(Pegawai peg) {
        int uang = peg.infoGaji();
        if (peg instanceof Manajer) {
            uang += ((Manajer) peg).infoTunjangan();
        } else if (peg instanceof Programmer) {
            uang += ((Programmer) peg).infoBonus();
        }
        return uang;
    }

    public static void main(String args[]) {
        Pegawai man = new Manajer("Agus", 800, 50);
        Pegawai prog = new Programmer("Budi", 600, 30);
        Bayaran hr = new Bayaran();

        System.out.println("Bayaran untuk Manajer : "
            + hr.hitungbayaran(man));
        System.out.println("Bayaran untuk Programmer : "
            + hr.hitungbayaran(prog));
    }
}
  
```

D. Tugas Rumah

Buat program Java yang mengimplementasikan konsep *polymorphism* dengan ketentuan program menerima masukan dari user dan program belum pernah dibuat pada praktikum sebelumnya.